Remote Control Method for Mobile Robot Based on Force Feedback Generated using Collision Prediction Map

Motoi, Naoki

Kobayashi, Masato

Masaki, Ryo

(Citation)
IEEJ Journal of Industry Applications,8(4):727-735

(Issue Date)
2019

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/90007924

IEEJ Journal of Industry Applications
Vol.8 No.4 pp.727–735 DOI: 10.1541/ieejjia.8.727

Paper

Remote Control Method for Mobile Robot Based on Force Feedback Generated using Collision Prediction Map

Naoki Motoi* Senior Member, Masato Kobayashi* Student Member
Ryo Masaki* Student Member

(Manuscript received Aug. 1, 2018, revised Dec. 12, 2018)

This paper proposes a remote control method for a mobile robot based on the force feedback generated using a collision prediction map. The collision prediction map expresses the relation between the mobile robot and its surrounding environment as the collision prediction time at each translational and angular velocity. The force feedback is generated by using this collision prediction map. In the proposed method, the operator can feel the environmental information as the tactile sensation. This improves the operability of the remote control system. The validity of the proposed method was confirmed from the experimental results.

Keywords: motion control, mobile robot, remote control, force feedback

1. Introduction

Japan has faced many natural disasters such as earthquakes and tsunami. From this background, several disaster defense systems by using robotics technology have been developed. Mobile robots to collect the environmental information in the disaster sites are treated as one of these systems [1][2]. Therefore, researches on mobile robots have been actively reported.

Focused on the researches on mobile robots, the control methods are roughly classified into two types; One is the autonomous control type, and the other is the remote control type. The autonomous mobile robots have achieved intelligent tasks by using several sensor information [3][4]. This technology is useful for the contribution of the labor force, since it is possible to conduct tasks without operators. Abbas *et al.* proposed the obstacle avoidance with the nonlinear model predictive control of the autonomous vehicle [5]. Lee *et al.* developed the efficient simultaneous localization and mapping (SLAM) using a monocular vision sensor for indoor service robots [6]. Hirose *et al.* reported the following control approach based on the model predictive control [7]. However, it is hard for these mobile robots to be applied in the complicated environment such as the disaster site.

On the other hand, there are also a lot of researches on remote control of mobile robots. By the assistance based on the operator's judgement, remote control systems can achieve several tasks in the complicated environment. Okura *et al.* proposed the free-viewpoint mobile robot teleoperation based on the vision sensor [8]. Zhao *et al.* developed the brain-machine interfacing-based teleoperation for the mobile robots [9]. Frank *et al.* reported the mobile mixed-reality approach to interact with and control a multi-robot system [10]. However, it is hard to recognize surrounding environment by using the only visual information. Therefore, the operator needs training for the skillful operation.

For the operability improvement, the remote control methods with the force feedback have been reported. In these control methods, the operator recognizes the surrounding environment by not only the visual information but also the tactile information [11][12]. Pecka *et al.* reported the tactile terrain exploration by using the mobile robot for traversing rough terrains [13]. Xu *et al.* presents the visual-haptic aid teleoperation system [14]. Ma and Ben-Tzvi developed the teleoperation with the haptic glove for the mobile robot navigation [15]. Masone *et al.* proposed the shared framework for the trajectory planning via the force-feedback haptic interface for the mobile robots [16]. However, most of these conventional methods do not consider the non-holonomic constraint. Therefore, the force commands that mobile robots do not realize may be generated. As a result, the operability of the remote control may not be improved. Considering the non-holonomic constraint, we proposed the force command generation method based on the translational and angular velocity dimension [17][18]. This research showed the improper force feedback did not improve the operability. On the other hand, the operability improvement was experimentally shown by using the proper force feedback at appropriate timing [18].

This paper focuses on the remote control method for the mobile robot with the force feedback. As with the methods [17][18], the force commands are generated based on the translational and angular velocity dimension. This paper proposes the collision prediction map which is calculated from the prediction trajectory and the environmental information. This map expresses the collision probability on several translational and angular velocities. By using this collision prediction map and the velocity commands, the force feedback depending on the collision probability is achieved. As a result, the operator can feel the environmental information as

* Graduate School of Maritime Sciences, Kobe University
5-1-1, Fukaeminami, Higashinada-ku, Kobe 658-0022, Japan

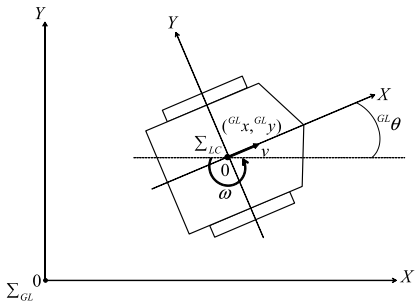


Fig. 1. Coordinate Systems

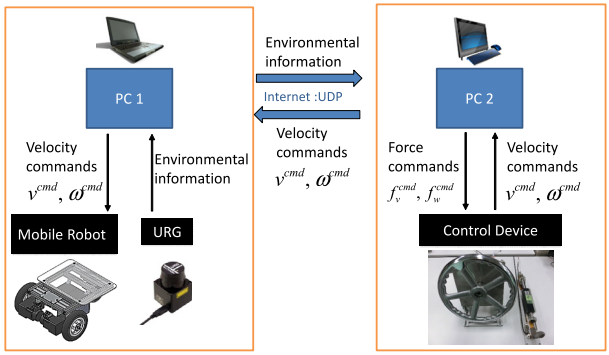


Fig. 2. Remote Control System with Force Feedback

the tactile sensation. Therefore, the operability for the remote control is improved. The validity of the proposed method was confirmed from the experimental results.

This paper is organized as follows. Section 2 describes the modeling of the mobile robot. In section 3, the remote control system with the force feedback is explained. Sections 4 and 5 describe the generation method of force commands as the conventional and proposed methods. In section 6, experimental results are shown to confirm the validity of the proposed remote control system. Finally, this paper is concluded in section 7.

2. Modeling

This section explains the modeling of the mobile robot. Figure 1 shows the coordinate systems. As shown in Fig. 1, the global coordinate system Σ_{GL} and the local coordinate system Σ_{LC} are defined. The origin of the global coordinate system is constant, and is defined as the center point of the robot at the initial position. On the other hand, the origin of the local coordinate system is set to the center point of the robot, and is changed during the moving. The directions of X-axis and Y-axis are defined as the same direction as the translational motion, and the vertical left direction of X-axis.

The relation between the velocities in the global coordinate system and these in the local coordinate system is expressed as follows.

$$ {}^{GL}\dot{x} = v\cos({}^{GL}\theta) \quad \cdots\cdots (1) $$
$$ {}^{GL}\dot{y} = v\sin({}^{GL}\theta) \quad \cdots\cdots (2) $$
$$ {}^{GL}\dot{\theta} = \omega \quad \cdots\cdots (3) $$

where x, y and θ represent the positon and posture of the mobile robot. The superscript ^{GL} means the value in the global coordinate system. On the other hand, the values in the local coordinate system do not use the superscript. v and ω stand for the translational and angular velocities.

3. Remote Control System

This section shows the remote control system for the mobile robot. Figure 2 shows the remote control system with the force feedback. This remote control system consists of the mobile robot and two motors as the control device. Two PCs are utilized to control the mobile robot and the control device. As the laser range finder, URG manufactured by HOKUYO AUTOMATIC CO., LTD. is used [19]. URG was attached to the mobile robot to measure the environmental information.

3.1 Velocity Commands This subsection describes the generation of velocity commands. One linear motor is used for the translational velocity command, and one rotary motor is utilized for the angular velocity command. The translational velocity command v^{cmd} and the angular velocity command ω^{cmd} are generated in proportional to the movement of each motor.

$$ v^{cmd} = V^{max} \cdot x_v^{res} / X^{max} \quad \cdots\cdots (4) $$
$$ \omega^{cmd} = \Omega^{max} \cdot \theta_\omega^{res} / \Theta^{max} \quad \cdots\cdots (5) $$

where V^{max} and Ω^{max} represent the maximum values of the translational and angular velocities. X^{max} and Θ^{max} mean the maximum displacement and maximum rotation angle of motors. These values are decided from the specifications of the mobile robot and motors. x_v^{res} and θ_ω^{res} stand for the position response of linear motor and the angle response of the rotary motor. The translational and angular velocity commands are sent to the mobile robot by UDP.

3.2 Force Controller This subsection shows the force feedback in the control device. For the acceleration control, disturbance observer (DOB) [21] is implemented in each motor. In addition, reaction force observer (RFOB) [22] is also implemented to estimate the reaction force without the force sensor. The force controller based on the acceleration control is expressed.

$$ \ddot{x}_v^{ref} = K_f(f_v^{cmd} - \hat{f}_v^{ext}) \quad \cdots\cdots (6) $$
$$ \ddot{\theta}_\omega^{ref} = K_f(f_\omega^{cmd} - \hat{f}_\omega^{ext}) \quad \cdots\cdots (7) $$

where K_f means the force feedback gain. f^{cmd} and $\hat{f}^{ext}$ represent the force command and reaction force estimated by RFOB. The superscript ^{ref} means the reference value. The subscripts ${}_v$ and ${}_\omega$ stand for the values related to the translational velocity and the angular velocity.

By using (6) and (7), the force commands are realized. Therefore, the operator can feel the force feedback as the tactile sensation. On the other hand, it is possible to manipulate the control device with the small operational force, if force commands are equal to 0 ($f^{cmd} = 0$). The generation methods of force commands are detailed in sections 4 and 5.

4. Conventional Method

Figure 3 shows the force command generation for the remote control as the conventional method [11][12]. In this method, the force command is generated in proportion to the distance between the obstacle and the mobile robot. The force command f^{cmd} is calculated as follows.

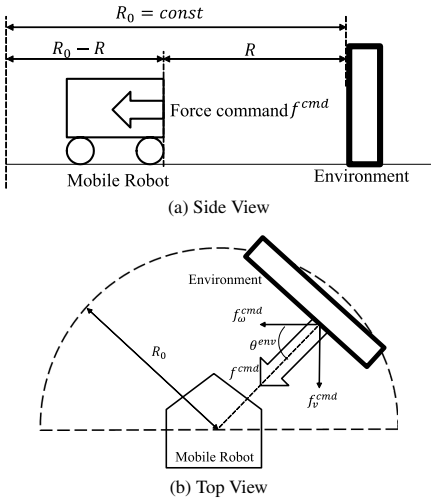


Fig. 3. Force Command Generation using Geometric Relation

$$f^{cmd} = \begin{cases} K_i \cdot (R_0 - R) & R_0 > R \\ 0 & otherwise \end{cases} \quad \cdots\cdots\cdots\cdots\cdots (8)$$

where R and R_0 stand for the distance from the mobile robot to the closest environment and the distance threshold. K_i is the variable feedback gain to generate the force command. As shown in Fig. 3(a), the force command is generated in the condition of $R_0 > R$.

The feedback gain K_i is modified due to the differentiation of the distance from the mobile robot to the closest environment.

$$K_i = \begin{cases} K^{min} & \frac{dR}{dt} \geq 0 \\ \frac{1}{\gamma}(K^{max} - K^{min})\frac{dR}{dt} + K^{min} & -\gamma < \frac{dR}{dt} < 0 \\ K^{max} & \frac{dR}{dt} \leq -\gamma \end{cases} \quad \cdots\cdots\cdots\cdots\cdots (9)$$

where K^{max} and K^{min} mean the maximum and minimum feedback gains. γ represents the velocity limitation.

In order to achieve this force command, it is necessary to divide the force command into the translational and angular directions. For this division, the angle between the mobile robot and closest environment as shown in Fig. 3(b) is utilized.

$$f_v^{cmd} = f^{cmd} \cdot \sin\theta^{env} \cdots\cdots\cdots\cdots\cdots (10)$$
$$f_\omega^{cmd} = f^{cmd} \cdot \cos\theta^{env} \cdots\cdots\cdots\cdots\cdots (11)$$

where θ^{env} represents the angle between the mobile robot and the closest environment.

5. Proposed Method

This section proposes the force generation method using the collision prediction map. Firstly, the collision prediction map is detailed. Secondly, the force commands are generated by the relation between the collision prediction map and velocity commands.

5.1 Collision Prediction Method This subsection explains the collision prediction method between the mobile robot and environment. This collision prediction is used to generate the force commands. For the collision prediction, dynamic window is calculated [4]. Dynamic window is composed of the translational and angular velocities which the mobile robot can realize after the unit time ΔT[s]. The unit time ΔT should be set considering the relation between the maximum/minimum velocities and maximum accelerations from the robot specification. Dynamic window is expressed as follows.

$$v_d^{max} = v^{res} + \dot{V}^{max} \cdot \Delta T \cdots\cdots\cdots\cdots\cdots (12)$$
$$v_d^{min} = v^{res} - \dot{V}^{max} \cdot \Delta T \cdots\cdots\cdots\cdots\cdots (13)$$
$$\omega_d^{max} = \omega^{res} + \dot{\Omega}^{max} \cdot \Delta T \cdots\cdots\cdots\cdots\cdots (14)$$
$$\omega_d^{min} = \omega^{res} - \dot{\Omega}^{max} \cdot \Delta T \cdots\cdots\cdots\cdots\cdots (15)$$

where v^{res} and ω^{res} mean the translational and angular velocity responses. These velocity responses are treated as the current velocities. The subscript $_d$ is the values related to dynamic window. The superscripts ^{max} and ^{min} represent the maximum and minimum values. V^{max} and Ω^{max} stand for the maximum translational and angular velocities. v_d^{max}, v_d^{min}, ω_d^{max} and ω_d^{min} are the maximum and minimum values of the translational and angular velocities after ΔT [s]. Considering the robot specification, dynamic window is modified.

$$v_d^{max} = \begin{cases} V^{max} & if\ v_d^{max} > V^{max} \\ v_d^{max} & otherwise \end{cases} \cdots\cdots\cdots\cdots\cdots (16)$$
$$v_d^{min} = \begin{cases} V^{min} & if\ v_d^{min} < V^{min} \\ v_d^{min} & otherwise \end{cases} \cdots\cdots\cdots\cdots\cdots (17)$$
$$\omega_d^{max} = \begin{cases} \Omega^{max} & if\ \omega_d^{max} > \Omega^{max} \\ \omega_d^{max} & otherwise \end{cases} \cdots\cdots\cdots\cdots\cdots (18)$$
$$\omega_d^{min} = \begin{cases} \Omega^{min} & if\ \omega_d^{min} < \Omega^{min} \\ \omega_d^{max} & otherwise \end{cases} \cdots\cdots\cdots\cdots\cdots (19)$$

where V^{min} and Ω^{min} mean the minimum translational and angular velocities from the robot specification. In this research, the minimum translational velocity is equal to 0 ($V^{min} = 0$). In other words, this research focuses on the forward motion. Dynamic window is expressed as the velocity area from v_d^{min} to v_d^{max} and from ω_d^{min} to ω_d^{max}.

In order to calculate the collision prediction, dynamic window is divided into N along the translational velocity and M along the angular velocity. N and M are natural numbers and are defined as follows.

$$N = \left\lfloor \frac{v_d^{max} - v_d^{min}}{\Delta V} \right\rfloor \cdots\cdots\cdots\cdots\cdots (20)$$
$$M = \left\lfloor \frac{\omega_d^{max} - \omega_d^{min}}{\Delta \Omega} \right\rfloor \cdots\cdots\cdots\cdots\cdots (21)$$

where ΔV and $\Delta\Omega$ represent the resolutions of the translational and angular velocities for the collision prediction. $\lfloor\ \rfloor$ means floor function. As a result, dynamic window is divided into $N \times M$. The higher the resolutions of velocities are, the more precise the collision prediction is calculated. However, the communication time delay between two PCs occurs, since the calculation cost and the data amount of the collision prediction are increased. Considering this tradeoff, it is necessary to set resolutions of the translational and angular velocities for the collision prediction.

Figure 4 shows the flowchart of the collision prediction between the robot and environment. As shown in Fig. 4, this flowchart consists of 6 steps.

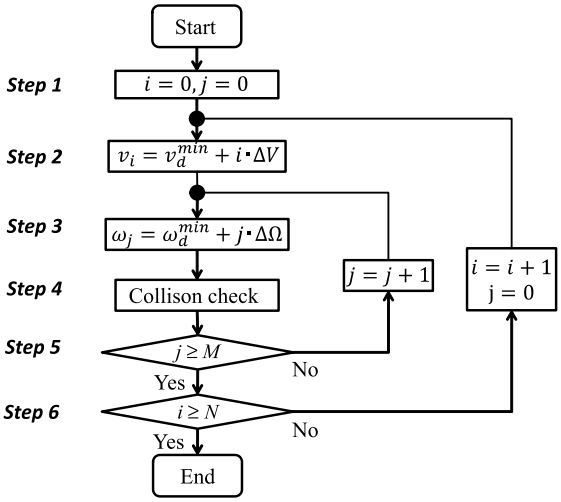


Fig. 4. Flowchart for Collision Check

Step 1: Coefficients $i(i = 0, 1, 2, \cdots, N)$ and $j(j = 0, 1, 2, \cdots, M)$ are initialized.

$$i = 0 \quad \cdots\cdots (22)$$

$$j = 0 \quad \cdots\cdots (23)$$

Step 2: The translational velocity for the collision check v_i is calculated.

$$v_i = v_d^{min} + i \cdot \Delta V \quad \cdots\cdots (24)$$

Step 3: The angular velocity for the collision check ω_j is calculated.

$$\omega_j = \omega_d^{min} + j \cdot \Delta\Omega \quad \cdots\cdots (25)$$

Step 4: The collision check is conducted by using the translational velocity v_i, the angular velocity ω_j and the environmental information measured by URG. The detail of the collision check is described in Section 5.2.

Step 5: If j is larger than M, go to *Step 6*. Otherwise, $j = j+1$ and go to *Step 3*.

Step 6: If i is larger than N, this flowchart is finished. Otherwise, $i = i + 1$, $j = 0$ and go to *Step 2*.

5.2 Collision Prediction Map This subsection proposes the collision prediction map. Figure 5 shows the flowchart to calculate the collision prediction map. This flowchart is inserted in *Step 4* of Fig. 4. As shown in Fig. 5, the collision check is conducted.

Step 4-1: By the relation between $|\omega_j|$ and Ω_{th}, the robot motion is divided into 2 cases. Ω_{th} is the threshold value in the angular velocity. In addition, Ω_{th} should be set to the value close to 0.

Step 4-2A: In the case of $|\omega_j| \leq \Omega_{th}$, the robot motion is treated as the straight motion as shown in Fig. 6(a). These environmental information measured by URG is expressed as $(x_l^s, y_l^s)(l = 1, 2, \cdots, P)$. P is the number of measurement points by URG. By using the prediction trajectory considering the robot width D, the environment where there is the collision possibility is extracted.

$$y_l^s > -\frac{D}{2} \quad \cdots\cdots (26)$$

$$y_l^s < \frac{D}{2} \quad \cdots\cdots (27)$$

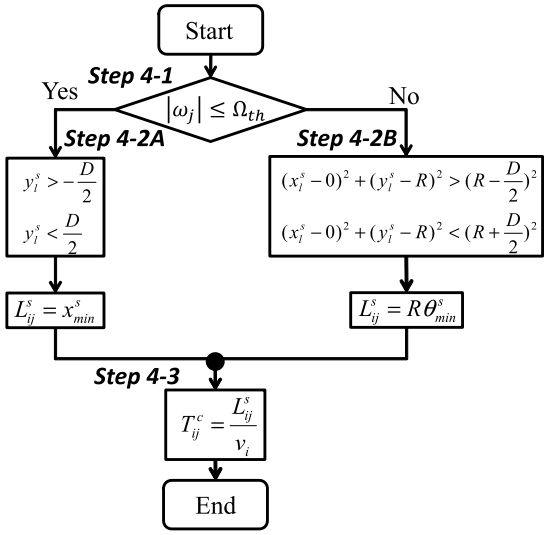
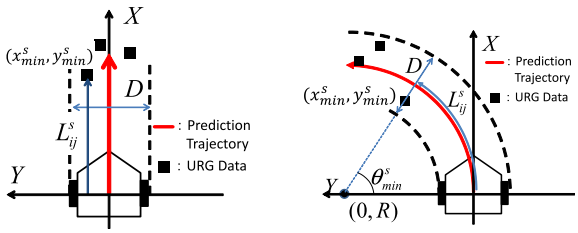


Fig. 5. Flowchart for Collision Prediction Map



(a) $|\omega_j| \leq \Omega_{th}$ (b) $|\omega_j| > \Omega_{th}$

Fig. 6. Calculation of Prediction Collision Time

The closest environment within (26) and (27) is defined as (x_{min}^s, y_{min}^s). The distance from the robot to (x_{min}^s, y_{min}^s) is calculated.

$$L_{ij}^s = x_{min}^s \quad \cdots\cdots (28)$$

where L_{ij}^s means the distance to the closest environment from the robot in the case of v_i and ω_j.

Step 4-2B: In the case of $|\omega_j| > \Omega_{th}$, the robot motion is treated as the circular movement. In this case, the turning radius R is calculated from the v_i and ω_j.

$$R = \frac{v_i}{\omega_j} \quad \cdots\cdots (29)$$

The center of the rotation is expressed as $(0, R)$ in the local coordinate system. Therefore, the environment of the high collision probability is extracted by using the prediction trajectory considering the robot width D.

$$(x_l^s - 0)^2 + (y_l^s - R)^2 > \left(R - \frac{D}{2}\right)^2 \quad \cdots\cdots (30)$$

$$(x_l^s - 0)^2 + (y_l^s - R)^2 < \left(R + \frac{D}{2}\right)^2 \quad \cdots\cdots (31)$$

In (30) and (31), the closest environment to the mobile robot is searched. This closest environment has the highest probability of collision with the robot. The angle between the robot and closest environment is defined as θ_{min}^s. The distance from the robot to the closest environment is calculated from the circumference.

$$L_{ij}^s = R\theta_{min}^s \quad \cdots\cdots (32)$$

Step 4-3: The prediction time T^c_{ij} to the collision between the robot and the environment is calculated.

$$T^c_{ij} = \frac{L^s_{ij}}{v_i} \cdots\cdots (33)$$

In (33), the prediction time in the case of v_i and ω_j is estimated. As shown in Fig. 4, this calculation for the collision prediction is conducted for $N \times M$ sets of translational and angular velocities. By using $N \times M$ sets of the prediction time, the collision prediction map is obtained.

5.3 Generation Method of Force Commands This subsection proposes the generation method of force commands by using the collision prediction map and velocity commands. The collision prediction map is composed of the prediction time calculated from v_i and ω_j. v_i and ω_j are the translational and angular velocities for the collision check, and these values are discrete values. On the other hand, the velocity commands v^{cmd} and ω^{cmd} are continues values. Therefore, the interpolation using Bezier surface is conducted.

The closest points of the velocity commands in the collision prediction map are calculated as follows.

$$n = \left\lfloor \frac{v^{cmd} - V^{min}_d}{\Delta V} \right\rfloor \cdots\cdots (34)$$

$$m = \left\lfloor \frac{\omega^{cmd} - \Omega^{min}_d}{\Delta \Omega} \right\rfloor \cdots\cdots (35)$$

where n and m mean the integral numbers. The collision prediction time for v^{cmd} and ω^{cmd} is obtained by the interpolation using Bezier surface. At first, the interpolation along the angular velocity is conducted.

$$\hat{t}^c_{v_{(n+k)}\omega^{cmd}} = a_k\omega_p^3 + b_k\omega_p^2 + c_k\omega_p + d_k \cdots\cdots (36)$$

where a_k, b_k, c_k, d_k $(k = -1, 0, 1)$ and ω_p are expressed as follows.

$$a_k = \frac{-T^c_{(n+k)(m+1)} + 2T^c_{(n+k)m} - T^c_{(n+k)(m-1)}}{4\Delta\Omega^3} \cdots\cdots (37)$$

$$b_k = \frac{3(T^c_{(n+k)(m+1)} - 2T^c_{(n+k)m} + T^c_{(n+k)(m-1)})}{4\Delta\Omega^2} \cdots\cdots (38)$$

$$c_k = \frac{T^c_{(n+k)(m+1)} - T^c_{(n+k)(m-1)}}{2\Delta\Omega} \cdots\cdots (39)$$

$$d_k = T^c_{(n+k)m} \cdots\cdots (40)$$

$$\omega_p = \omega^{cmd} - \omega_m \cdots\cdots (41)$$

where $\hat{t}^c_{v_{(n+k)}\omega^{cmd}}$ means the collision prediction time in the case of $v_{(n+k)}$ and ω^{cmd}.

By using (36), the interpolation along the translational velocity is calculated.

$$\hat{t}^c_{v^{cmd}\omega^{cmd}} = ev_p^3 + fv_p^2 + gv_p + h \cdots\cdots (42)$$

where e, f, g, h and v_p are described as follows.

$$e = \frac{-T^c_{(n+1)m} + 2T^c_{nm} - T^c_{(n-1)m}}{4\Delta V^3} \cdots\cdots (43)$$

$$f = \frac{3(T^c_{(n+1)m} - 2T^c_{nm} + T^c_{(n-1)m})}{4\Delta V^2} \cdots\cdots (44)$$

$$g = \frac{T^c_{(n+1)m} - T^c_{(n-1)m}}{2\Delta V} \cdots\cdots (45)$$

$$h = T^c_{nm} \cdots\cdots (46)$$

$$v_p = v^{cmd} - v_n \cdots\cdots (47)$$

where $\hat{t}^c_{v^{cmd}\omega^{cmd}}$ means the collision prediction time in the case of v^{cmd} and ω^{cmd}.

By using the collision prediction time $\hat{t}^c_{v^{cmd}\omega^{cmd}}$, the force commands are generated.

$$f^{cmd} = \begin{cases} \alpha(\hat{t}^c_{v^{cmd}\omega^{cmd}} - T_{th}) & if\ \hat{t}^c_{v^{cmd}\omega^{cmd}} < T_{th} \\ 0.0 & otherwise \end{cases} \cdots\cdots (48)$$

where T_{th} sands for time threshold. α means the coefficient to generate the force commands. If the collision prediction time $\hat{t}^c_{v^{cmd}\omega^{cmd}}$ is smaller than time threshold T_{th}, the collision probability is high. Therefore, the force commands generated. In the case of $\hat{t}^c_{v^{cmd}\omega^{cmd}} > T_{th}$, the collision probability is low. In this case, the force command is set to 0, and the operator manipulates the control device with the small manipulation force.

The coefficient of the force command α should be decided considering the relation between the maximum force of the control device and the time threshold. By setting the time threshold T_{th} high, the safety remote control is achieved. On the other, the frequent force commands occur depending on the environment condition such as the narrow space. This frequent force commands may not connect the operability improvement. From this viewpoint, time threshold should be decided considering the environment condition and the operability of the remote control system.

The control device consists of the linear motor for the translational direction and the rotary motor for the angular direction as shown in Fig. 2. For the realization of the force feedback in the control device, it is necessary to divide the force command into the translational direction and the angular direction. Considering the gradients of the collision prediction map with respect to the translational and angular velocities, it is possible to recover the collision prediction time by the force feedback. As a result, the force command is divided into the translational direction and the angular direction by the gradient in the collision prediction map.

$$f^{cmd}_v = f^{cmd}\frac{\Delta t_v}{\sqrt{\Delta t_v{}^2 + \Delta t_\omega{}^2}} \cdots\cdots (49)$$

$$f^{cmd}_\omega = f^{cmd}\frac{\Delta t_\omega}{\sqrt{\Delta t_v{}^2 + \Delta t_\omega{}^2}} \cdots\cdots (50)$$

$$\Delta t_v = \frac{\partial \hat{t}^c_{v^{cmd}\omega^{cmd}}}{\partial v} \cdots\cdots (51)$$

$$\Delta t_\omega = \frac{\partial \hat{t}^c_{v^{cmd}\omega^{cmd}}}{\partial \omega} \cdots\cdots (52)$$

where Δt_v and Δt_ω mean the gradients with respect to the translational and angular velocities. By substituting (49) and (50) for (6) and (7), the operator can feel collision probability as the tactile sensation.

6. Experiments

This section confirmed the validity of the proposed method from the experimental results. Firstly, the experimental setup is explained. Secondly, the experimental results are shown.

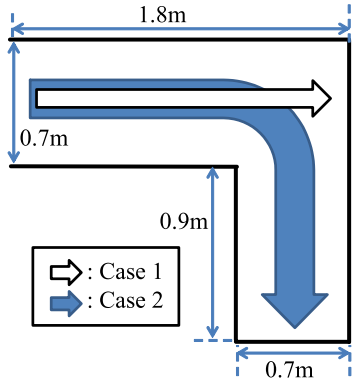


Fig. 7. Experimental Course

Table 1. Specification of Mobile Robot

V^{max}	Maximum translational velocity	0.4 [m/s]
V^{min}	Minimum translational velocity	0.0 [m/s]
Ω^{max}	Maximum angular velocity	1.5 [m/s]
Ω^{min}	Minimum angular velocity	-1.5 [m/s]
$\dot{V}^{max}$	Maximum acceleration	1.0 [m/s^2]
$\dot{\Omega}^{max}$	Maximum angular acceleration	2.0 [m/s^2]
D	Mobile robot width	0.4 [m]

Table 2. Control Parameters

ΔT	Unit time for prediction map	1.0 [s]
ΔV	Resolution of translational velocity	0.05 [m/s]
$\Delta\Omega$	Resolution of angular velocity	0.1 [rad/s]
Ω_{th}	Angular velocity threshold	0.05 [rad/s]
K_f	Force feedback gain	1.0

Table 3. Parameters for Force Command

R_0	Distance threshold	0.5 [m]
K^{min}	Minimum feedback gain	20.0
K^{max}	Maximum feedback gain	40.0
γ	Velocity limitation	0.15 [m/s]
α	Coefficient of force command	6.0
T_{th}	Time threshold in Case 1	4.5 [s]
T_{th}	Time threshold in Case 2	2.5 [s]

6.1 Experimental Setup This subsection describes the experimental setup. Figure 2 shows the remote control system which was utilized in this experiment. This mobile robot is T-frog project i-Cart mini [20]. Tables 1–3 show the specification of the mobile robot, control parameters, and parameters for the force command. The force feedback gain K_f, parameters in the conventional method, R_0, K^{min}, K^{max}, and γ are set by the trials and errors in this paper.

The experiments are divided into 2 cases.

Case 1: The mobile robot moved along the straight line as shown in Fig. 7. In Case 1, the comparison between the conventional method and the proposed method was evaluated. For the evaluation under the same condition, the only constant translational velocity command was set (v^{cmd} = 0.15, ω^{cmd} = 0.0). In addition, this experiment did not conduct the force feedback, but the generation of the force commands. From the difference of the force commands, the validity of the force command generation was checked.

Case 2: The mobile robot moved along the right turned course as shown in Fig. 7. In Case 2, the proposed method was implemented. From this experiment, the validity of the proposed remote control system with the force feedback was confirmed.

6.2 Experimental Results This subsection shows the experimental results to confirm the validity of the proposed method. Figures 8-9 show the trajectory of the mobile robot, and experimental results in Case 1. In Fig. 8, the robot position was calculated from the odometry information. URG data was used as the environmental information. As shown in Fig. 8, the mobile robot moved along the straight line by the remote operation. Fig. 10 shows the collision prediction map at points 1-4 shown in Fig. 8. The collision prediction map was changed due to the relation between the mobile robot and the environmental information.

Figure 9(a), (b) and (c) represent the velocity commands, the force commands by the conventional method and proposed method. In Case 1, the only constant translational

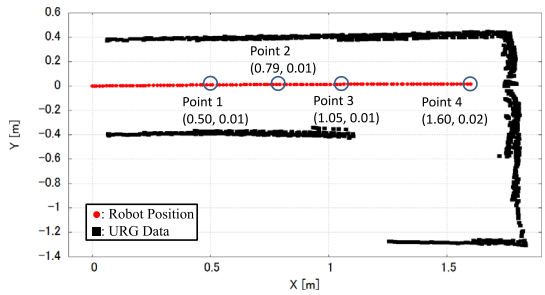
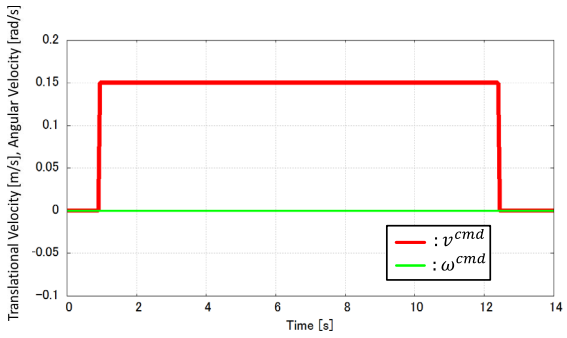
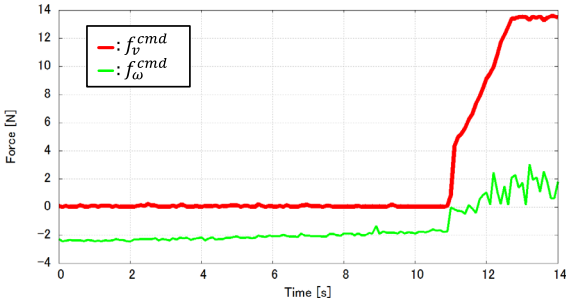


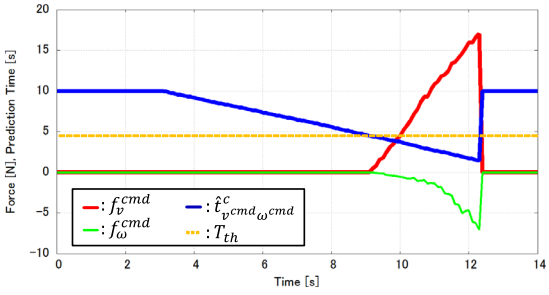
Fig. 8. Trajectory of Mobile Robot (Case 1)



(a) Velocity Commands

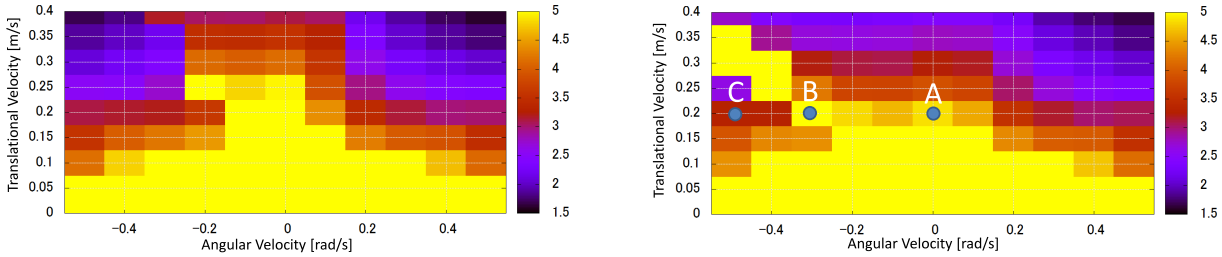


(b) Force Commands (Conventional Method)



(c) Force Commands (Proposed Method)

Fig. 9. Experimental Results (Case 1)



(a) Point 1 (t = 6.0 [s])　(b) Point 2 (t = 7.5 [s])

(c) Point 3 (t = 9.0 [s])　(d) Point 4 (t = 13.0 [s])

Fig. 10.　Collision Prediction Map (Case 1)

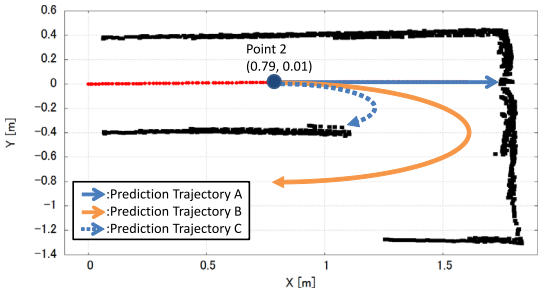


Fig. 11.　Prediction Trajectory from Point 2

Table 4.　Collision Prediction Time

Point	T_{ij}^c [s]	v_i [m/s]	ω_j [rad/s]	L_{ij}^s [m]
A	4.86	0.2	0.0	0.97 by (28)
B	5.0	0.2	-0.3	No Collision by (32)
C	3.30	0.2	-0.5	0.66 by (32)

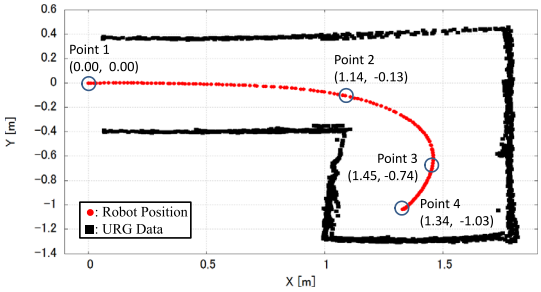
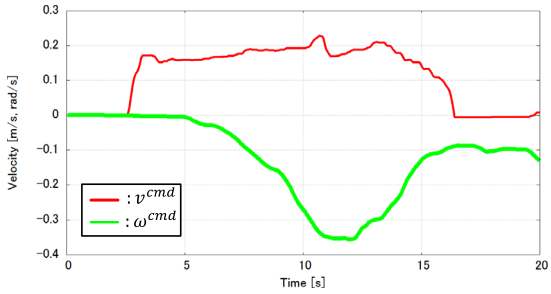
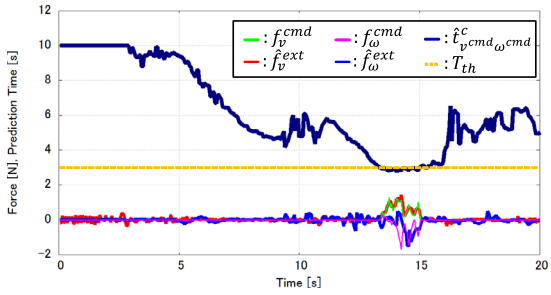


Fig. 12.　Trajectory of Mobile Robot (Case 2)



(a) Velocity Commands

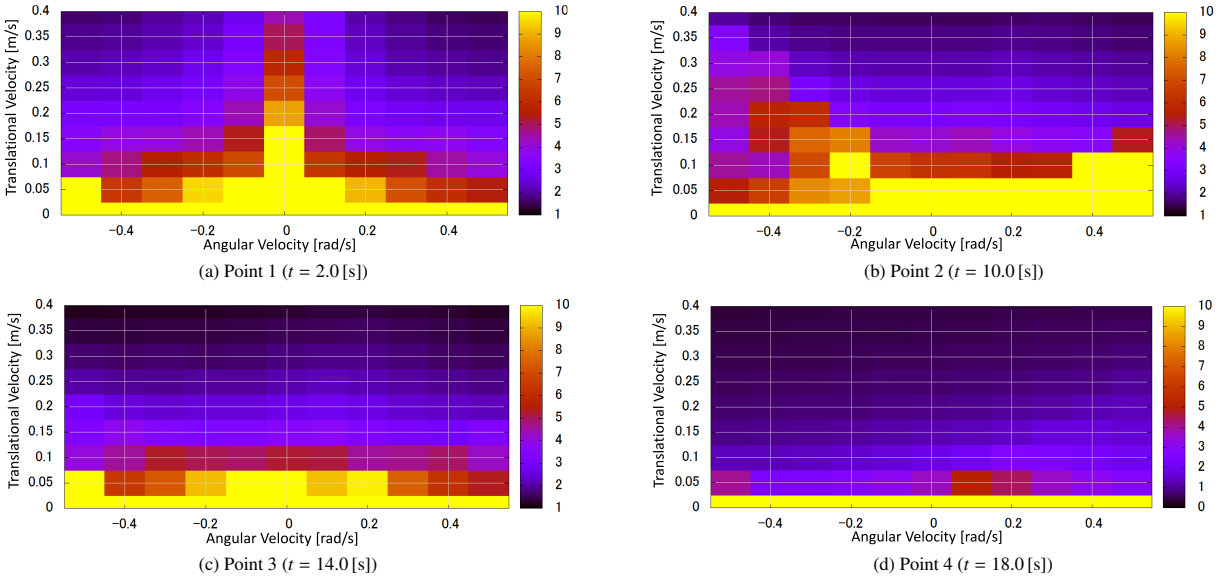


(b) Force Responses

Fig. 13.　Experimental Results (Case 2)

velocity command was sent from the operation side as shown in Fig. 9(a). In the conventional method, the force command was generated due to the geometric relation between the mobile robot and environment. The force commands were generated from the start point to the goal point as shown in Fig. 9(b), though there is no risk to collide with the side environment. The force command may be treated as the heavy manipulation force. Therefore, the operability may not be improved. On the other hand, the proposed method generated the force command by using the collision prediction map as shown in Fig. 9(c). When the collision probability is high, the operator grasped the environment information by the force feedback. In addition, the operator manipulated the control device with the small operational force in the case of low collision probability.

The accuracy of the collision prediction map was checked at points A-C in Fig. 10(b). For the check of the collision prediction time, Fig. 11 shows the prediction trajectories from point 2. Table 4 shows the accuracy of the collision prediction map. In Case 1, the maximum collision prediction time is set to 5.0 [s]. Therefore, the collision prediction time at point B is equal to 5.0 [s], since the prediction trajectory B did not collide with the environment. On the other hand, the prediction trajectories A and C collide with the environment. As shown in Table 4, the collision prediction time T_{ij}^c was accurately calculated from the geometric relation. From



(a) Point 1 (t = 2.0 [s])　(b) Point 2 (t = 10.0 [s])

(c) Point 3 (t = 14.0 [s])　(d) Point 4 (t = 18.0 [s])

Fig. 14. Collision Prediction Map (Case 2)

these results, the reliability of the collision prediction map was confirmed.

Figure 12 shows the trajectory of the mobile robot in Case 2. Figure 13(a) and (b) represent the velocity commands, and the force responses by the proposed method. In the proposed remote control system, the operator manipulated the control device while watching the visual information and feeling the tactile information. According to the operator's manipulation, the velocity commands in Case 2 were generated by (4), (5). By the velocity commands as shown in Fig. 13(a), the mobile robot moved along the desired right turned trajectory as shown in Fig. 12.

Figure 14 shows the collision prediction map at points 1–4. The maximum collision prediction time was set to 10.0 [s] in Case 2. In the case of high collision prediction, the force commands were generated as shown in Fig. 13(b). As a result of this force feedback, the velocity commands were modified to avoid the collision. In addition, the proposed method conducted the more delicate force feedback compared with the conventional methods. Therefore, the operability improvement is expected [18].

From these experimental results, the validity of the proposed method was confirmed.

7. Conclusions

This paper proposed the remote control method for the mobile robot by using the force feedback. The force commands are calculated from the relation between the collision prediction map and velocity commands. The collision prediction map is generated from the collision estimation between the robot and environment. By using the proposed method, it is possible to realize the force feedback depending on the collision probability. As a result, the operator grasps the environment information as the tactile sensation. In addition, the operator manipulates the control device with the small operational force in the case of low collision probability. From these experimental results, the validity of the proposed method was confirmed.

The future works are described as follows.

- The design method of controller parameters considering the operability should be developed.
- The experiments under the dynamic environment should be conducted.
- The quantitative proof method to evaluate the operability should be developed.

Acknowledgment

This research was partially supported by JSPS KAKENHI (19K04454), Nagamori Foundation, and International Maritime Research Center of Kobe University.

References

(1) K. Tadakuma, E. Takane, M. Fujita, A. Nomura, H. Komatsu, M. Konyo, and S. Tadokoro: "Planar Omnidirectional Crawler Mobile Mechanism—Development of Actual Mechanical Prototype and Basic Experiments", *IEEE Robotics and Automation Letters*, Vol.3, No.1, pp.395–402 (2018)

(2) M. Schwarz, T. Rodehutskors, D. Droeschel, M. Beul, M. Schreiber, N. Araslanov, I. Ivanov, C. Lenz, J. Razlaw, S. Schller, D. Schwarz, A.T. Kyniazopoulou, and S. Behnke: "NimbRo Rescue: Solving Disaster-response Tasks with the Mobile Manipulator Robot Momaro", *Journal of Field Robotics*, Vol.34, No.2, pp.400–425 (2017)

(3) Y. Hasegawa and Y. Fujimoto: "Experimental Verification of Path Planning with SLAM", *IEEJ Journal of Industry Applications*, Vol.5, No.3, pp.253–260 (2016)

(4) D. Fox, W. Burgard, and S. Thrun: "The Dynamic Window Approach to Collision Avoidance", *Proceedings of IEEE International Conference on Robotics & Automation Magazine*, Vol.4, No.1, pp.23–33 (2005)

(5) M.A. Abbas, R. Milman, and J.M. Eklund: "Obstacle Avoidance in Real Time With Nonlinear Model Predictive Control of Autonomous Vehicles", *Canadian Journal of Electrical and Computer Engineering*, Vol.40, No.1, pp.12–22 (2017)

(6) T. Lee, C. Kim, and D.D. Cho: "A Monocular Vision Sensor-Based Efficient SLAM Method for Indoor Service Robots", *IEEE Transactions on Industrial Electronics*, Vol.66, No.1, pp.318–328 (2019)

(7) N. Hirose, R. Tajima, N. Koyama, K. Sukigara, and M. Tanaka: "Following Control Approach Based on Model Predictive Control for Wheeled Inverted Pendulum Robot", *Advanced Robotics*, Vol.30, No.6, pp.374–385 (2016)

(8) F. Okura, Y. Ueda, T. Sato, and N. Yokoya: "Free-viewpoint Mobile Robot Teleoperation Interface Using View-dependent Geometry and Texture", *ITE Transactions on Media Technology and Applications*, Vol.2, No.1, pp.82–93 (2014)

(9) S. Zhao, Z. Li, R. Cui, Y. Kang, F. Sun, and R. Song: "BrainMachine Interfacing-Based Teleoperation of Multiple Coordinated Mobile Robots", *IEEE Transactions on Industrial Electronics*, Vol.64, No.6, pp.5161–5170 (2017)

(10) J.A. Frank, S.P. Krishnamoorthy, and V. Kapila: "Toward Mobile Mixed-Reality Interaction With Multi-Robot Systems", *IEEE Robotics and Automation Letters*, Vol.2, No.4, pp.1901–1908 (2017)

(11) I. Farkhatdinov, J.H. Ryu, and J. Poduraev: "Rendering of Environmental Force Feedback in Mobile Robot Teleoperation based on Fuzzy Logic", *Proceedings of IEEE International Symposium on Computational Intelligent in Robotics and Automation*, pp.503–508 (2009)

(12) I. Farkhatdinov, J.H. Ryu, and J. Poduraev: "A Preliminary Experimental Study on Haptic Teleoperation of Mobile Robot with Variable Force Feedback Gain", *Proceedings of IEEE International Symposium on Haptics*, pp.251–256 (2010)

(13) M. Pecka, K. Zimmermann, M. Reinstein, and T. Svoboda: "Controlling Robot Morphology From Incomplete Measurements", *IEEE Transactions on Industrial Electronics*, Vol.64, No.2, pp.1773–1782 (2017)

(14) X. Xu, A. Song, D. Ni, H. Li, P. Xiong, and C. Zhu: "Visual-Haptic Aid Teleoperation Based on 3-D Environment Modeling and Updating", *IEEE Transactions on Industrial Electronics*, Vol.63, No.10, pp.6419–6428 (2016)

(15) Z. Ma and P. Ben-Tzvi: "RML Glove—An Exoskeleton Glove Mechanism With Haptics Feedback", *IEEE Transactions on Mechatronics*, Vol.20, No.2, pp.641–652 (2015)

(16) C. Masone, M. Mohammadi, P.R. Giordano, and A. Franchi: "Shared planning and control for mobile robots with integral haptic feedback", *The International Journal of Robotics Research*, Vol.37, No.11, pp.1395–1420 (2018)

(17) N. Motoi and H. Kimura: "Remote Control Method for Mobile Robot with Virtual Force Feedback Based on Environmental Information", *Proceedings of IEEJ International Workshop on Sensing, Actuation, and Motion Control*, IS3-3, pp.1–6 (2016)

(18) N. Motoi, H. Kimura, and M. Kobayashi: "Experimental Operability Evaluation of Remote Control with Force Feedback for Mobile Robot", *Proceedings of IEEE International Conference on Industrial Technology*, pp.159–164 (2018)

(19) HOKUYO AUTOMATIC CO., LTD., http://www.hokuyo-aut.co.jp/

(20) T-frog Project, http://t-frog.com/

(21) K. Ohnishi, M. Shibata, and T. Murakami: "Motion Control for Advanced Mechatronics", *IEEE/ASME Transactions on Mechatronics*, Vol.1, No.1, pp.56–67 (1996)

(22) T. Murakami, F. Yu, and K. Ohnishi: "Torque Sensorless Control in Multidegree-of-Freedom Manipulator", *IEEE/ASME Transactions on Industrial Electronics*, Vol.40, No.2, pp.259–265 (1993)

Naoki Motoi (Senior Member) received the B.E. degree in system design engineering and the M.E. and Ph.D. degrees in integrated design engineering from Keio University, Japan, in 2005, 2007 and 2010, respectively. In 2007, he joined the Partner Robot Div., Toyota Motor Corporation, Japan. From 2011 to 2013, he was a research associate at Yokohama National University. Since 2014, he has been with Kobe University, Japan, where he is a currently Associate Professor. His research interests include robotics, motion control, and haptic.

Masato Kobayashi (Student Member) received the B.E. degree in marine engineering from Kobe University, Japan, in 2017. He is currently working towards the M.E. degree in the graduate school of maritime sciences, Kobe University, Japan. His research interests include robotics, and motion control.

Ryo Masaki (Student Member) received the B.E. degree in marine engineering from Kobe University, Japan, in 2018. He is currently working towards the M.E. degree in the graduate school of maritime sciences, Kobe University, Japan. His research interests include haptics, and motion control.